The purpose of this project was to predict daily weather metrics for a given zip code based on historical weather data scraped from the web. <http://www.wunderground.com> was used to gather the historical data. The data provided by the website has readings taken roughly every hour for the following variables: Date, Time, Temperature, Dew Point, Humidity, Sea Level Pressure, Visibility, Wind Direction, Wind Speed, Gust Speed, Precipitation, Events, and Conditions. Data from 10/31/2010 to 10/31/2016 was gathered for my zip code (60502), and a zip code to the southwest (61350), as weather generally tends to travel Northeast in the northern hemisphere. The predictions of the target variables were made for my zip code, but both its historical data and the data from the zip code to the southwest were used as input variables to the models. Below is a sample of what the raw data look like:

**Figure** **1**: Snippet of raw weather data for zip code 60502 on 12/8/2016

To avoid high variance, only variables believed to be good predictors of the target variables were kept. Daily weighted averages (based on time) were then calculated for each variable. Lastly, values for yesterday’s weather metrics and average values for the previous week’s weather metrics were calculated for both zip codes. The conditions column had to be codified, i.e. assigned a numeric variable in place of the character variables when being used for prediction. The idea here was that one day alone (yesterday) couldn’t provide enough prediction value, given that weather is rather volatile in nature. To counter this, several days of weather leading up to the target date (today) were included. Once the preprocessing was complete, the final dataset had the following columns, where “Target” denotes variables being predicted and “Feature” denotes predictors of the target variables:

|  |  |
| --- | --- |
| **Variable Name** | **Variable Type** |
| 60502\_TemperatureF\_today | Target |
| 60502\_PrecipitationIn\_today | Target |
| 60502\_Conditions\_today | Target |
| 60502\_TemperatureF\_yesterday | Feature |
| 60502\_PrecipitationIn\_yesterday | Feature |
| 60502\_Conditions\_yesterday | Feature |
| 60502\_Dew.PointF\_yesterday | Feature |
| 60502\_Sea.Level.PressureIn\_yesterday | Feature |
| 60502\_Wind.SpeedMPH\_yesterday | Feature |
| 60502\_Humidity\_yesterday | Feature |
| 60502\_TemperatureF\_previous\_week | Feature |
| 60502\_PrecipitationIn\_ previous\_week | Feature |
| 60502\_Conditions\_ previous\_week | Feature |
| 60502\_Dew.PointF\_ previous\_week | Feature |
| 60502\_Sea.Level.PressureIn\_ previous\_week | Feature |
| 60502\_Wind.SpeedMPH\_ previous\_week | Feature |
| 60502\_Humidity\_previous\_week | Feature |
| 61350\_TemperatureF\_yesterday | Feature |
| 61350\_PrecipitationIn\_yesterday | Feature |
| 61350\_Conditions\_yesterday | Feature |
| 61350\_Dew.PointF\_yesterday | Feature |
| 61350\_Sea.Level.PressureIn\_yesterday | Feature |
| 61350\_Wind.SpeedMPH\_yesterday | Feature |
| 61350\_Humidity\_yesterday | Feature |
| 61350\_TemperatureF\_previous\_week | Feature |
| 61350\_PrecipitationIn\_ previous\_week | Feature |
| 61350\_Conditions\_ previous\_week | Feature |
| 61350\_Dew.PointF\_ previous\_week | Feature |
| 61350\_Sea.Level.PressureIn\_ previous\_week | Feature |
| 61350\_Wind.SpeedMPH\_ previous\_week | Feature |
| 61350\_Humidity\_previous\_week | Feature |
| Month | Feature |

**Figure 2**: List of columns in dataset

There were a few different prediction tasks being done. Today’s temperature and today’s precipitation are both continuous variables, whereas conditions is a discrete variables with values such as “Clear”, “Partly Cloudy”, “Rain”, etc. I thought it would be interesting to build models for these variables because they provide a wide variety of prediction tasks to perform. Predicting temperature is a rather traditional regression mode, where the range of outputs varies a fair amount. Predicting precipitation could be a regression model, as the outcome is continuous, but the majority of days in the zip code used in this analysis have values of 0 inches of precipitation for the day. With that in mind, precipitation outcomes was changed to a Boolean variable, being assigned a value of 1 if precipitation was present for a day and a value of 0 if it was not present, making it a classification problem. Lastly, predicting conditions was a rather typical classification problem using supervised learning, where 10 different outcomes were possible. The outcomes included unknown, clear, partly cloudy, mostly cloudy, fog, light rain/drizzle, rainy, thunderstorms/heavy rain, snow, and freezing rain/sleet. The below paragraphs detail the methods, results, and interpretations of each prediction task. In all cases, 70% of the data was used in the training set and the remainder was used in the test set. With more time, validation sets would have been created to tune these models as well.

***Task 1: Predicting Temperature***

Three different regression models were used to predict temperature. These models included support vector machine, logistic regression, and random forest. The below table displays some evaluation metrics describing how the models performed on the test set:

**Figure 3**: Evaluation metrics of temperature prediction regression models

The first thing that sticks out immediately is the apparent poor performance of a couple of the models when looking at R2 and RMSE. However, the median absolute error is reasonable. Digging into the predictions, this is due to a couple of temperature predictions that are quite high – over 1,000 degrees Fahrenheit. While many of the predictions are not bad, these couple of predictions over 1,000 degrees Fahrenheit are obviously problematic. I’m not quite sure what would have caused this behavior, but I speculate that

***Task 2: Predicting Precipitation***

Three different classification models were used to predict the presence of precipitation. These models included k-nearest neighbors, Naïve Bayes, and a random forest. The below table displays some evaluation metrics describing how the models performed on the test set:

**Figure 4**: Evaluation metrics of precipitation prediction classification models

And below are the ROC curves

***Task 3: Predicting Conditions***

Three different classification models were used to predict the day’s conditions. These models included k-nearest neighbors, Naïve Bayes, and a random forest. The below table displays some evaluation metrics describing how the models performed on the test set:

**Figure 5**: Evaluation metrics of conditions prediction classification models

Overall, while these models are far from perfect, it proved to be an interesting exercise in solving a variety of prediction problems. This really illustrated the necessity of trying several different models, as the performance can widely vary among them. As mentioned earlier, it would be more ideal to try even more models, spend time tuning the models, and even spend more time working on variable selection and bringing in more variables.